APL

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/755,621	01/02/2001	Jeffrey T. Tuatini	243768001US1	2988

25096 7590 05/28/2004

PERKINS COIE LLP
PATENT-SEA
P.O. BOX 1247
SEATTLE, WA 98111-1247

EXAMINER	
CHOUDHURY, AZIZUL Q	
ART UNIT	PAPER NUMBER
2143	8

DATE MAILED: 05/28/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PRG

Office Action Summary	**Application No.** 09/755,621	**Applicant(s)** TUATINI, JEFFREY T.	
	Examiner Azizul Choudhury	**Art Unit** 2143	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 02 January 2001.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-43 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-43 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 02 July 2001 is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date 7.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

Detailed Action

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-43 are rejected under 35 U.S.C. 102(b) as being anticipated by Leach et al (US Pat No: US006108715A), hereafter referred to as Leach.

1. With regards to claim 1, Leach teaches a method in a computer system for executing applications to request services from remote shared service providers each able to provide services to multiple executing applications, each remote shared service provider having an associated distinct proxy component for communicating with the remote shared service provider, comprising: under control of a first of the executing applications, instantiating a local messaging component within the first application, the local messaging component able to communicate with each of the associated proxies; notifying the local messaging component to request a specified service from a remote shared service provider identified with a unique name; and under control of the local messaging component, requesting the specified service by, determining based on the unique name identified in the notification to request the specified service from a first of

the remote shared service providers; retrieving configuration information for the first remote shared service provider that identifies a first proxy component associated with the first remote shared service provider; when the first proxy component is not already instantiated within the first application, instantiating the first proxy component within the first application; sending a message to the first proxy component requesting that the first remote shared service provider provide the specified service; and receiving a response from the first remote shared service provider via the first proxy component, and notifying the first application of the received response, so that an application can communicate with any of the remote shared service providers via a local messaging component that interacts with a proxy component associated with the remote shared service provider (Leach teaches a design for RPC (remote procedure calls). This design allows for the remote invoking of procedures from a client machine on a server machine (column 3, lines 14-20, Leach). A client machine sends a request with parameters and the server serves the request and provides a result to the client. To successfully perform these tasks, the claimed features such as "unique name" and "messaging" inherently must occur. For instance, Leach's system makes requests; this process is equivalent to messaging (column 3, lines 14-51, Leach). Furthermore, proxies are used within Leach's design (column 4, line 61, Leach)).

2. With regards to claim 2, Leach teaches a method including, under control of the first application: notifying the local messaging component to request a second specified service from a remote shared service provider identified with a second unique name;

and under control of the local messaging component, requesting the second specified service by, determining based on the second unique name identified in the notification to request the second specified service from a second of the remote shared service providers; retrieving configuration information for the second remote shared service provider that identifies a second proxy component associated with the second remote shared service provider; sending a message to the second proxy component requesting that the second remote shared service provider provide the second specified service; and receiving a response from the second remote shared service provider via the second proxy component, and notifying the first application of the received response (Leach's design allows for remote procedure calls. Leach's design uses signatures along with addresses to uniquely identify each remote process request and result (column 3, lines 14-51, Leach). This shows that multiple remote process requests are allowed within Leach's design).

3. With regards to claim 3, Leach teaches a method including: under control of a second of the executing applications, instantiating a second local messaging component within the second application, the second local messaging component able to communicate with each of the associated proxies; notifying the second local messaging component to request a specified service from a remote shared service provider identified with the unique name of the first shared service provider; and under control of the second local messaging component, requesting the specified service by, determining based on the unique name identified in the notification to request the

specified service from the first remote shared service provider; retrieving the configuration information for the first remote shared service provider that identifies the associated first proxy component; when the first proxy component is not already instantiated within the second application, instantiating the first proxy component within the second application; and sending a message to the first proxy component requesting that the first remote shared service provider provide the specified service (Leach's design allows for remote procedure calls. Leach's design uses signatures along with addresses to uniquely identify each remote process request and result (column 3, lines 14-51, Leach). This shows that multiple remote process requests are allowed within Leach's design. In addition, "messaging" inherently must occur. For instance, Leach's system makes requests, this process is equivalent to messaging (column 3, lines 14-51, Leach). Furthermore, proxies are used within Leach's design (column 4, line 61, Leach)).

4. With regards to claim 4, Leach teaches a method including, under control of the first proxy component: receiving the sent message from the local messaging component; identifying based on the received message one of multiple services provided by the first remote shared service provider that is requested; requesting the first remote shared service provider to provide the identified service; receiving a response from the first remote shared service provider; and sending the received response to the local messaging component (Leach's design allows for remote procedure calls. The design allows for parameters to allow for the client's request

details to be sent along with the process request to the server (column 3, lines 18-22, Leach). In addition, Leach states how the means exist by which to identify the appropriate server processes (column 4, lines 43-48, Leach)).

5. With regards to claim 5, Leach teaches a method including, under control of the first remote shared service provider: receiving the request from the first proxy component to provide the identified service; performing processing associated with the identified service, the processing generating a response; and sending the generated response to the first proxy component (Leach's design allows for the client to make a process request to the server, the server then processes the request and generates a response back to the client (column 3, lines 14-51, Leach). In addition, Leach allows for the use of proxies (column 4, line 61, Leach)).

6. With regards to claim 6, Leach teaches a method wherein the first remote shared service provider has multiple associated proxy components each for communicating with the first remote shared service provider to request distinct services provided by the first remote shared service provider, wherein the notifying of the local messaging component further identifies a name of the specified service, and wherein the identifying of the first proxy component from the retrieved configuration information is based at least in part on the identified name of the specified service (Leach's design allows for the client to make a process request to the server, the server then processes the request and generates a response back to the client (column 3, lines 14-51, Leach). In

addition, Leach allows for the use of proxies (column 4, line 61, Leach). Furthermore, Leach's design allows for other identifying means such as signatures to ensure the proper process is being requested at the proper time and parameters to ensure that the server uses the correct application for the process (column 3, lines 14-51, Leach)).

7. With regards to claim 7, Leach teaches a method wherein the local messaging component and the first proxy component are software modules, and wherein the instantiating of the components 3 within the first application includes executing the software modules (Leach's design allows for remote procedure calls. Leach's design uses signatures along with addresses to uniquely identify each remote process request and result (column 3, lines 14-51, Leach). This shows that multiple remote process requests are allowed within Leach's design. In addition, "messaging" inherently must occur. For instance, Leach's system makes requests, this process is equivalent to messaging (column 3, lines 14-51, Leach). Furthermore, proxies are used within Leach's design (column 4, line 61, Leach)).

8. With regards to claim 8, Leach teaches a method wherein the local messaging component and the first proxy component are instances of object classes, and wherein the instantiating of the components within the first application includes creating the instances (It is inherent that when components are instantiated, that they have to be created).

9. With regards to claim 9, Leach teaches a method wherein the notifying of the local messaging component includes invoking an interface method of the local messaging component instance (It is inherent that when a messaging component is used, an instance of it must first be invoked).

10. With regards to claim 10, Leach teaches a method wherein the sending of the message to the first proxy component includes invoking an interface method of the first proxy component instance (It is inherent that when a proxy component is to be used, an interface must first be invoked).

11. With regards to claim 11, Leach teaches a method wherein the configuration information is retrieved from storage via a directory service using the unique name of the first remote shared service provider (Leach's design allows for the process parameters (details) to be stored. First, there are stacks within the client and the server (column 3, lines 35-39, Leach). Second there is are tables that hold data that identify processes (column 4, line 45, Leach)).

12. With regards to claim 12, Leach teaches a method including, before the executing of the first application, storing the configuration information using the directory service (Before processes are run, parameters are stored within the stacks (column 3, lines 35-39, Leach)).

13. With regards to claim 13, Leach teaches a method wherein the retrieved configuration information includes initialization information for the first proxy component, and wherein the instantiating of the first proxy component includes providing the initialization information to a the first proxy component for use as part of its initialization (The proxies handle the parameters (information) as claimed within Leach's design (column 4, line 58 – column 5, line 15, Leach)).

14. With regards to claim 14, Leach teaches a method wherein the notifying of the request of the specified service includes a message to be sent to the first remote shared service provider, and wherein the message sent to the first proxy component is the included message (The request notification must be sent to the service provider (server). This is inherent and occurs within Leach's design (column 4, line 58 – column 5, line 15, Leach)).

15. With regards to claim 15, Leach teaches a method wherein the notifying of the request of the specified service includes a message to be sent to the first remote shared service provider, and wherein the message sent to the first proxy component is a version of the included message that is transformed under the control of the local messaging component before the sending (The request notification must be sent to the service provider (server). This is inherent and occurs within Leach's design (column 4, line 58 – column 5, line 15, Leach). It is further inherent that since the messenger

sends the message (request), that the message will be transformed into a format that can be handled by the messenger).

16. With regards to claim 16, Leach teaches a method wherein the first application includes multiple action handler components each for responding to requests from external clients with responses generated in a format used by the action handler component, and wherein the notifying of the request of the specified service is prompted by a request generated by one of the action handler components (The request notification must be sent to the service provider (server). This is inherent and occurs within Leach's design (column 4, line 58 – column 5, line 15, Leach). It is further inherent that since the messenger sends the message (request), that the message will be transformed into a format that can be handled by the messenger).

17. With regards to claim 17, Leach teaches a method wherein the first application includes multiple view handler components each for transforming generated responses in formats used by action handler components into formats compatible with external clients, and including, after the notifying of the received response from the first remote shared service provider, using one of the view handler components to transform the received response into a format compatible with an external client (The request notification must be sent to the service provider (server). This is inherent and occurs within Leach's design (column 4, line 58 – column 5, line 15, Leach). It is further inherent that since the messenger sends the message (request), that the message will

be transformed into a format that can be handled by the messenger. Leach, also discloses that the results are sent from the server to the client (column 3, lines 40-51, Leach). Such a process is the same as the client sending a message (request) to the server, except the direction changes. Hence, the claimed formatting and handlers still exist within Leach's design).

18. With regards to claim 18, Leach teaches a computer-readable medium whose contents cause a computing device to execute applications able to request services from remote shared service providers each able to provide services to multiple executing applications, each remote shared service provider having an associated proxy component for communicating with the remote shared set-vice provider, by: under control of a first of the executing applications, instantiating a local messaging component, the local messaging component able to communicate with each of the associated proxies; notifying the local messaging component to request a specified service from a remote shared service provider; and under control of the local messaging component, requesting the specified service by, determining based on the notifying to request the specified service from a first of the remote shared service providers; retrieving configuration information for the first remote shared service provider that identifies a first proxy component associated with the first remote shared service provider; when the first proxy component is not already instantiated, instantiating the first proxy component; sending a message to the first proxy component requesting that the first remote shared service provider provide the specified service; and receiving a

response from the first remote shared service provider via the first proxy component (Leach teaches a design for RPC (remote procedure calls). This design allows for the remote invoking of procedures from a client machine on a server machine (column 3, lines 14-20, Leach). A client machine sends a request with parameters and the server serves the request and provides a result to the client. To successfully perform these tasks, the claimed features such as "unique name" and "messaging" inherently must occur. For instance, Leach's system makes requests, this process is equivalent to messaging (column 3, lines 14-51, Leach). Furthermore, proxies are used within Leach's design (column 4, line 61, Leach)).

19. With regard to claim 19, Leach teaches a computing device for executing applications to request services from remote shared service providers each able to provide services to multiple executing applications, each remote shared service provider having an associated proxy component for communicating with the remote shared service provider, comprising: a first application capable of, during execution, instantiating a local messaging component able to communicate with each of the associated proxies and of notifying the local messaging component to request a specified service from a remote shared service provider; and the local messaging component capable of, during execution, determining to based on the notification to request the specified service from a first of the remote shared service providers, of retrieving configuration information for the first remote shared service provider that identifies a first proxy component associated with the first remote shared service

provider, of instantiating the first proxy component, of sending a message to the first proxy component requesting that the first remote shared service provider provide the specified service, and of receiving a response from the first remote shared service provider via the first proxy component (Leach teaches a design for RPC (remote procedure calls). This design allows for the remote invoking of procedures from a client machine on a server machine (column 3, lines 14-20, Leach). A client machine sends a request with parameters and the server serves the request and provides a result to the client. To successfully perform these tasks, the claimed features such as "unique name" and "messaging" inherently must occur. For instance, Leach's system makes requests, this process is equivalent to messaging (column 3, lines 14-51, Leach). Furthermore, proxies are used within Leach's design (column 4, line 61, Leach)).

20. With regards to claim 20, Leach teaches a method in a computer system for an executing application to communicate with remote services via a local messaging service, each of the remote services having an associated proxy for communicating with the remote service, the method comprising: for each of multiple of the remote services, notifying the local messaging service to send a message to the remote service; under control of the local messaging service, identifying the proxy associated with the remote service by retrieving information associated with the remote service; and sending the message to the identified proxy for communication to the remote service; and receiving a response to the sent message from the remote service via the identified proxy (Leach teaches a design for RPC (remote procedure calls). This design allows for the remote

invoking of procedures from a client machine on a server machine (column 3, lines 14-20, Leach). A client machine sends a request with parameters and the server serves the request and provides a result to the client. To successfully perform these tasks, the claimed features such as "unique name" and "messaging" inherently must occur. For instance, Leach's system makes requests, this process is equivalent to messaging (column 3, lines 14-51, Leach). Furthermore, proxies are used within Leach's design (column 4, line 61, Leach)).

21. With regards to claim 21, Leach teaches a method including, under control of each of the identified proxies: receiving the sent message from the local messaging service that is to be communicated to a remote service; sending the received message to the remote service; receiving a response from the remote service; and sending the received response to the local messaging service (Leach's design allows for proxies (column 4, line 61, Leach). In addition, messages (requests) are handled by the proxies between the client and server to perform the remote processes (column 4, line 58 – column 5, line 15, Leach)).

22. With regards to claim 22, Leach teaches a method including, before the sending of the received message to the remote service, transforming the received message into a format appropriate for the remote service (The request notification must be sent to the service provider (server). This is inherent and occurs within Leach's design (column 4, line 58 – column 5, line 15, Leach). It is further inherent that since the messenger

sends the message (request), that the message will be transformed into a format that can be handled by the messenger).

23. With regards to claim 23, Leach teaches a method wherein each of the remote services has multiple associated proxies, wherein the notifying of the local messaging service identifies a manner of sending the message to the remote service, and wherein the identifying of the associated proxy from the retrieved information includes selecting one of the multiple associated proxies based at least in part on the identified manner (Leach's design allows for the client to make a process request to the server; the server then processes the request and generates a response back to the client (column 3, lines 14-51, Leach). In addition, Leach allows for the use of proxies (column 4, line 61, Leach). Furthermore, Leach's design allows for other identifying means such as signatures to ensure the proper process is being requested at the proper time and parameters to ensure that the server uses the correct application for the process (column 3, lines 14-51, Leach)).

24. With regards to claim 24, Leach teaches a method wherein the local messaging service and the proxies are created objects, and including creating the objects (Leach's design has objects created (column 4, lines 58-65, Leach)).

25. With regards to claim 25, Leach teaches a method wherein the information associated with a remote service is retrieved from a directory service using a unique

name of the remote service (Leach's design allows for signatures for procedures to be retrieved from a table (column 3, lines 40-45, Leach). In addition, Leach teaches that the stacks keep maintain copies of signatures at both the client and the server (column 3, lines 35-40, Leach)).

26. With regards to claim 26, Leach teaches a method including storing the associated information with the directory service before the retrieving (Leach's design has the parameters for the process to be performed, pushed onto a stack before performing the operation (column 4, lines 58-65, Leach)).

27. With regards to claim 27, Leach teaches a method wherein the application includes multiple action handlers each able to respond to requests from external clients with responses generated in a format used by the action handler, and wherein the notifying of the local messaging service to send the messages is prompted by requests generated by the action handlers (The request notification must be sent to the service provider (server). This is inherent and occurs within Leach's design (column 4, line 58 – column 5, line 15, Leach). It is further inherent that since the messenger sends the message (request), that the message will be transformed into a format that can be handled by the messenger. Leach, also discloses that the results are sent from the server to the client (column 3, lines 40-51, Leach). Such a process is the same as the client sending a message (request) to the server, except the direction changes. Hence, the claimed formatting and handlers still exist within Leach's design).

28. With regards to claim 28, Leach teaches a method wherein the application includes multiple view handlers each able to transform a generated response in a format used by an action handler into a format compatible with at least one of the external clients, and including, after the receiving of the response from the remote service, using one of the view handlers to transform the received response into a format compatible with an external client (The request notification must be sent to the service provider (server). This is inherent and occurs within Leach's design (column 4, line 58 – column 5, line 15, Leach). It is further inherent that since the messenger sends the message (request), that the message will be transformed into a format that can be handled by the messenger. Leach, also discloses that the results are sent from the server to the client (column 3, lines 40-51, Leach). Such a process is the same as the client sending a message (request) to the server, except the direction changes. Hence, the claimed formatting and handlers still exist within Leach's design).

29. With regards to claim 29, Leach teaches a computer-readable medium whose contents cause a computing device to execute an application able to communicate with remote services via a local messaging service, each of the remote services having an associated proxy for communicating with the remote service, by: for each of multiple of the remote services, notifying the local messaging service to send a message to the remote service; under control of the local messaging service, identifying the proxy associated with the remote service by retrieving information associated with the remote

service; and sending the message to the identified proxy for communication to the remote service; and receiving a response to the sent message from the remote service via the identified proxy (Leach teaches a design for RPC (remote procedure calls). This design allows for the remote invoking of procedures from a client machine on a server machine (column 3, lines 14-20, Leach). A client machine sends a request with parameters and the server serves the request and provides a result to the client. To successfully perform these tasks, the claimed features such as "unique name" and "messaging" inherently must occur. For instance, Leach's system makes requests, this process is equivalent to messaging (column 3, lines 14-51, Leach). Furthermore, proxies are used within Leach's design (column 4, line 61, Leach)).

30. With regards to claim 30, Leach teaches a method wherein the computer readable medium is a data transmission medium transmitting a generated data signal containing the contents (A computer readable medium is simply a medium in which digital data can be stored. Both network mediums (such as Ethernet cabling) and computer memory (such as hard drives and RAM) are examples of computer readable mediums).

31. With regards to claim 31, Leach teaches a method wherein the computer readable medium is a memory of a computer system (A computer readable medium is simply a medium in which digital data can be stored. Both network mediums (such as

Ethernet cabling) and computer memory (such as hard drives and RAM) are examples of computer readable mediums).

32. With regards to claim 32, Leach teaches a computing device for executing an application to communicate with remote services via a local messaging service, each of the remote services having an associated proxy for communicating with the remote service, comprising: an application capable of, for each of multiple of the remote services, notifying the local messaging service to send a message to the remote service and receiving a response to the sent message from the remote service via a proxy associated with the remote service; and the local messaging service capable of identifying the proxy associated with the remote service by retrieving information associated with the remote service and of sending the message to the identified proxy for communication to the remote service (Leach teaches a design for RPC (remote procedure calls). This design allows for the remote invoking of procedures from a client machine on a server machine (column 3, lines 14-20, Leach). A client machine sends a request with parameters and the server serves the request and provides a result to the client. To successfully perform these tasks, the claimed features such as "unique name" and "messaging" inherently must occur. For instance, Leach's system makes requests; this process is equivalent to messaging (column 3, lines 14-51, Leach). Furthermore, proxies are used within Leach's design (column 4, line 61, Leach)).

33. With regards to claim 33, Leach teaches a computer system for executing an application to communicate with remote services via a local messaging service, each of the remote services having an associated proxy for communicating with the remote service, comprising: means for, for each of multiple of the remote services, notifying the local messaging service to send a message to the remote service and receiving a response to the sent message from the remote service via a proxy associated with the remote service; and means for identifying the proxy associated with the remote service by retrieving information associated with the remote service and sending the message to the identified proxy for communication to the remote service (Leach teaches a design for RPC (remote procedure calls). This design allows for the remote invoking of procedures from a client machine on a server machine (column 3, lines 14-20, Leach). A client machine sends a request with parameters and the server serves the request and provides a result to the client. To successfully perform these tasks, the claimed features such as "unique name" and "messaging" inherently must occur. For instance, Leach's system makes requests; this process is equivalent to messaging (column 3, lines 14-51, Leach). Furthermore, proxies are used within Leach's design (column 4, line 61, Leach)).

34. With regards to claim 34, Leach teaches a method in a computer system for invoking functions of remote applications, each of the remote applications having an associated proxy for communicating with the remote application, the method comprising: for each of multiple of the remote applications, receiving a request to invoke

a specified function of the remote application using at least one specified parameter value; retrieving information associated with the remote application that identifies the proxy associated with the remote application; when a copy of the identified proxy has not already been instantiated, instantiating a copy of the identified proxy; notifying the identified proxy copy of the specified function and the specified parameter values; under control of the identified proxy copy, invoking the specified function of the remote application using the specified parameter values; and receiving a response from the invoking of the specified function (Leach teaches a design for RPC (remote procedure calls). This design allows for the remote invoking of procedures from a client machine on a server machine (column 3, lines 14-20, Leach). A client machine sends a request with parameters and the server serves the request and provides a result to the client. To successfully perform these tasks, the claimed features such as "unique name" and "messaging" inherently must occur. For instance, Leach's system makes requests, this process is equivalent to messaging (column 3, lines 14-51, Leach). Furthermore, proxies are used within Leach's design (column 4, line 61, Leach)).

35. With regards to claim 35, Leach teaches a method wherein each of the remote applications has multiple associated proxies for invoking distinct functions of the remote application, and wherein the identifying of the proxy associated with the remote application from the retrieved information is based at least in part on the specified function (Leach's design allows for the client to make a process request to the server, the server then processes the request and generates a response back to the client

(column 3, lines 14-51, Leach). In addition, Leach allows for the use of proxies (column 4, line 61, Leach). Furthermore, Leach's design allows for other identifying means such as signatures to ensure the proper process is being requested at the proper time and parameters to ensure that the server uses the correct application for the process (column 3, lines 14-51, Leach)).

36. With regards to claim 36, Leach teaches a method wherein the proxies are object instances, and wherein the instantiating of the proxies includes creating the object instances (Leach's design has objects created (column 4, lines 58-65, Leach)).

37. With regards to claim 37, Leach teaches a method wherein the information is retrieved from a directory service using an identified name of the remote application (Leach's design allows for signatures for procedures to be retrieved from a table (column 3, lines 40-45, Leach). In addition, Leach teaches that the stacks keep maintain copies of signatures at both the client and the server (column 3, lines 35-40, Leach)).

38. With regards to claim 38, Leach teaches a method wherein the retrieved information includes initialization information for the identified proxy, and wherein the instantiating of the proxy includes providing the initialization information to the proxy for use during initialization (The proxies handle the parameters (information) as claimed within Leach's design (column 4, line 58 – column 5, line 15, Leach)).

39. With regards to claim 39, Leach teaches a method wherein the application includes multiple action handlers each able to respond to requests from external clients with responses generated in a format used by the action handler, and wherein the request to invoke the specified function is received from one of the action handler components (The request notification must be sent to the service provider (server). This is inherent and occurs within Leach's design (column 4, line 58 – column 5, line 15, Leach). It is further inherent that since the messenger sends the message (request), that the message will be transformed into a format that can be handled by the messenger. Leach, also discloses that the results are sent from the server to the client (column 3, lines 40-51, Leach). Such a process is the same as the client sending a message (request) to the server, except the direction changes. Hence, the claimed formatting and handlers still exist within Leach's design).

40. With regards to claim 40, Leach teaches a method wherein the application includes multiple view handlers each able to transform responses in formats used by action handlers into formats compatible with external clients, and including, after the receiving of the response from the invoking of the specified function, using one of the view handler components to transform the received response into a format compatible with an external client (The request notification must be sent to the service provider (server). This is inherent and occurs within Leach's design (column 4, line 58 – column 5, line 15, Leach). It is further inherent that since the messenger sends the message

(request), that the message will be transformed into a format that can be handled by the messenger. Leach, also discloses that the results are sent from the server to the client (column 3, lines 40-51, Leach). Such a process is the same as the client sending a message (request) to the server, except the direction changes. Hence, the claimed formatting and handlers still exist within Leach's design).

41. With regards to claim 41, Leach teaches a computer-readable medium whose contents cause a computing device to invoke functions of remote applications each having an associated proxy for communicating with the remote application, by: for each of multiple of the remote applications, receiving a request to invoke a specified function of the remote application using at least one specified parameter value; retrieving information associated with the remote application that identifies the proxy associated with the remote application; when a copy of the identified proxy has not already been instantiated, to instantiating a copy of the identified proxy; notifying the identified proxy copy of the specified function and the specified parameter values; under control of the identified proxy copy, invoking the specified function of the remote application using the specified parameter values; and is receiving a response from the invoking of the specified function (Leach teaches a design for RPC (remote procedure calls). This design allows for the remote invoking of procedures from a client machine on a server machine (column 3, lines 14-20, Leach). A client machine sends a request with parameters and the server serves the request and provides a result to the client. To successfully perform these tasks, the claimed features such as "unique name" and

"messaging" inherently must occur. For instance, Leach's system makes requests; this process is equivalent to messaging (column 3, lines 14-51, Leach). Furthermore, proxies are used within Leach's design (column 4, line 61, Leach)).

42. With regards to claim 42, Leach teaches a computing device for invoking functions of remote applications each 2 having an associated proxy for communicating with the remote application, comprising: a first component capable of, for each of multiple of the remote applications, receiving a request to invoke a specified function of the remote application using at least one s specified parameter value, retrieving information associated with the remote application that identifies the proxy associated with the remote application, instantiating a copy of the identified proxy, notifying the identified proxy copy of the specified function and the 8 specified parameter values, and receiving a response from the invoking of the specified function; and to multiple proxy copies each capable of receiving a notification of a specified function and specified parameter values for one of the remote applications and of invoking the specified function of the remote application using the specified parameter values (Leach teaches a design for RPC (remote procedure calls). This design allows for the remote invoking of procedures from a client machine on a server machine (column 3, lines 14-20, Leach). A client machine sends a request with parameters and the server serves the request and provides a result to the client. To successfully perform these tasks, the claimed features such as "unique name" and "messaging" inherently must occur. For instance, Leach's system makes requests; this process is equivalent to messaging

(column 3, lines 14-51, Leach). Furthermore, proxies are used within Leach's design (column 4, line 61, Leach)).

43. With regards to claim 43, Leach teaches a method in a computer system for components of an executing application to communicate with rernote services, each of the remote services having an associated proxy for communicating with the remote service, the executing application having a messaging service component for communicating with the associated proxies, having multiple action handler components each for responding to requests with responses generated in a format used by the action handler component, and having multiple view handler components each for transforming generated responses in formats used by action handler components into formats compatible with external clients, the method comprising: under control of each of multiple of the action handler components, receiving a request from a remote client to provide a service; and providing the requested service by, determining a need for functionality available from one of the remote services; notifying the local messaging component to request the needed functionality from the one remote service; receiving a response from the one remote service via the local messing component; generating a response in a format used by the action handler component; determining a view handler component able to transform the generated response from the format used by the action handler component to a format compatible with the remote client; and providing the generated response to the determined view handler component for the transforming; under control of the local messaging component, receiving notifications from multiple

action handler components each to request functionality from a remote service; and for each received notification, identifying the proxy associated with the remote service; and sending a request for the functionality to the identified proxy for communication to the remote service; receiving a response to the sent request from the remote service via the identified proxy; and notifying the action handler component of the received response; and under control of each of multiple of the view handler components, receiving responses generated by at least one action handler component in a format used by the action handler component; and for each received response, transforming the received response into a format that is compatible with the remote client for whom the received response was generated (Leach teaches a design for RPC (remote procedure calls). This design allows for the remote invoking of procedures from a client machine on a server machine (column 3, lines 14-20, Leach). A client machine sends a request with parameters and the server serves the request and provides a result to the client. To successfully perform these tasks, the claimed features such as “unique name” and “messaging” inherently must occur. For instance, Leach’s system makes requests; this process is equivalent to messaging (column 3, lines 14-51, Leach). Furthermore, proxies are used within Leach’s design (column 4, line 61, Leach)).

Remarks

The examiner has evaluated the application and has determined that the claims fail to describe the invention as being unique. In particular, the claims describe the traits and steps possessed within RPC (remote procedure call) designs. While the claims are lengthy, the examiner feels that the current claims are lacking substance. The descriptions the claims provide are simply that of most RPC designs. Should the applicants feel that there are certain features that would allow the claims to describe a design that is unique, they are encouraged to make such necessary changes to the claim language.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Azizul Choudhury whose telephone number is 703-305-7209. The examiner can normally be reached on M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, David Wiley can be reached on 703-308-5221. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AC

DAVID WILEY
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100